CLAIMS

What is claimed is:

1. A system for kernel-mode shimming comprising:

a plurality of driver components;

a common shim component that provides added functionality to the plurality of driver components; and

a context component associated with each driver component that retrieves and maintains driver context information.

2. The system of claim 1, wherein driver context information includes a driver's linkage configuration.

3. The system of claim 2, wherein the context component comprises a hook component that retrieves an address associated with a kernel- mode service from a driver component's import address table.

4. The system of claim 3, wherein the hook component determines the address of the context component.

5. The system of claim 4, the context component further comprising a thunk component that replaces at least one address associated with a kernel-mode service in the driver's import address table with the address of the context component so as to redirect the flow of execution from the kernel-mode service to the context component.

6. The system of claim 5, wherein the thunk component links the context component to the shim component.

7. The system of claim 6, wherein the thunk component provides the shim component with context information regarding the kernel-mode service replaced by the context component.

8. The system of claim 7, wherein shim component provides a link to the kernel-mode service to direct the flow of execution from the shim component to the service.

9. The system of claim 1, wherein the added functionality provided by the shim component includes compensating for a driver fault.

10. The system of claim 1, wherein the added functionality provided by the shim component includes providing diagnostic testing.

11. A system for shimming kernel-mode drivers comprising:

a driver loader component that loads drivers and generates a notification signal to indicate that a particular driver has been loaded;

a shim database that stores shim components, identify drivers to be shimmed, and associates one or more shim components with drivers to be shimmed; and

a shim engine component that receives a notification signal from the driver loader component, queries the shim database to determine if the particular loaded driver needs to be shimmed, and loads shim components associated with the driver.

12. The system of claim 11, wherein the shim engine is kernel-mode service.

13. The system of claim 11, wherein the shim engine generates a context component associated with a particular loaded driver the context component comprising:

a data structure identifying a kernel-mode procedure utilized by the driver; and

a thunk component for linking the driver to the context component and the context component to a shim component.

14. The system of claim 11, further comprising a diagnostic component for determining the cause of a system problem, instability or inefficiency and initiating corrective action.

15. The system of claim 14, wherein the corrective action includes locating and applying one or more shim components stored in the shim database to a driver.

16. The system of claim 14, wherein the corrective action includes notifying a user.

17. The system of claim 14, further comprising an interface component to facilitate development and deployment of a remedial shim component.

18. The system of claim 17, wherein the interface component includes a shim wizard that navigates a user through a series of steps to develop a shim component or apply a previously developed shim component to a driver.

19. A system for shimming kernel-mode drivers comprising:
a multitude of driver components;
a shim component common to the multitude of driver components; and
a means for providing driver unique context data to the shim component such that a shim component can identify its caller in an overall driver linkage configuration.

20. The system of claim 19, wherein the context data includes information regarding a kernel-mode procedure utilized by the driver.

21. A method for shimming a kernel-mode driver comprising:
generating a shim component common to several drivers;
generating driver unique context data associated with each driver to be shimmed;
providing the driver unique context data to the shim component such that the shim component can determine its caller in an overall driver linkage configuration.

22. The method of claim 19, wherein the caller is a driver.

23. The method of claim 21, wherein providing context data to the shim component includes passing the context data *via* a procedure or method parameter.

24. The method of claim 20, wherein providing context data to the shim component includes loading the data into memory.

25. The method of claim 20, further comprising storing the shim component and context data in a shim database in a manner that preserves the association between a shim component, context data, and a driver.

26. A computer readable medium having stored thereon computer executable instructions for carry out the method of claim 20.

27. A method for modifying kernel-mode drivers calls comprising:

receiving a signal indicating that a driver has been loaded;

querying a shim database to determine if the loaded driver has shim components associated therewith;

loading any shim components associated with the driver;

initializing a unique context for the driver; and

redirecting the driver to the shim component, wherein the unique context identifies the driver to the shim component.

28. The method of claim 27, wherein redirecting the driver to the shim component comprises replacing a driver import address table entry specifying a kernel-mode procedure to be imported with a pointer to the shim component.

29. The method of claim 28, further comprising calling the kernel-mode procedure replaced by the pointer to the shim component from the shim component.

30. A computer readable medium having stored thereon computer executable instructions for carry out the method of claim 27.